

ANALISIS PERBANDINGAN AKURASI DAN WAKTU PROSES ALGORITMA STEMMING ARIFIN-SETIONO DAN NAZIEF-ADRIANI PADA DOKUMEN TEKS BAHASA INDONESIA

Ardiles Sinaga¹⁾ dan Sahat Pandapotan Nainggolan²⁾

^{1,2}Teknologi Rekayasa Perangkat Lunak, Fakultas Vokasi, Institut Teknologi Del

^{1,2}Jl. Jl. Sisingamangaraja, Sitoluama, 22381

E-mail : ardiles.sinaga@del.ac.id¹⁾, sahat.nainggolan@del.ac.id²⁾

ABSTRAK

Information Retrieval adalah suatu sistem untuk menemukan informasi yang diinginkan oleh seorang pengguna pada sekumpulan informasi yang dinginkannya berdasarkan data-data yang diinputkan oleh pengguna lain sebelumnya (*query*). *Information Retrieval* merupakan suatu sistem untuk melakukan pencarian, penyimpanan, dan juga pemeliharaan informasi. Salah satu tahapan penting yang terdapat di dalam suatu *information retrieval* adalah proses *stemming*. *Stemming* adalah suatu rangkaian tindakan untuk memperoleh kata dasar (*root* atau *stem*) dari masing-masing kata dengan mengeliminasi seluruh imbuhan (*affixes*) yang terdapat dalam kata tersebut baik itu berupa sisipan (*suffixes*), awalan (*prefixes*), serta kombinasi antara awalan dan akhiran (*confixes*) pada kata turunan. Setiap algoritma *stemming* mempunyai kelebihan dan kekurangan. Secara umum, untuk mengukur keefektifan dari sebuah algoritma *stemming* dapat didasarkan pada berbagai parameter yaitu keakuratan, kecepatan proses, dan juga kesalahan. Penelitian ini membandingkan dua algoritma *stemming* yaitu algoritma Nazief Adriani dan algoritma Arifin Setiono untuk mengukur performansi dari masing-masing algoritma dengan melakukan pengujian sebanyak 30 dokumen teks berbahasa Indonesia. Adapun hasil pengujian pada penelitian ini memperlihatkan bahwa perbandingan performansi algoritma Nazief Adriani lebih baik dari algoritma Arifin Setiono di mana nilai rata-rata akurasi tertinggi dimiliki oleh algoritma Nazief Adriani yakni sebesar 97.73% dengan rata-rata waktu proses *stemming* selama 20.17 detik. Sedangkan algoritma Arifin Setiono memiliki nilai rata-rata akurasi sebesar 94.37% dengan rata-rata waktu proses *stemming* selama 23.32 detik.

Kata Kunci: *Information Retrieval, Stemming, Perbandingan, Akurasi, Waktu Proses, Nazief Adriani, Arifin Setiono*

1. PENDAHULUAN

Pencarian suatu informasi berbasis digital saat ini tidak bisa dipungkiri bahwa itu sudah menjadi kebutuhan untuk setiap orang. Seiring meningkatnya penggunaan teknologi informasi ditambah dengan peningkatan arus informasi yang begitu pesat, membuat pertumbuhan informasi yang tersedia secara digital juga ikut meningkat secara eksponensial. Informasi yang umumnya dicari adalah berupa sebuah dokumen. Untuk memperoleh sebuah dokumen yang dibutuhkan, pengguna biasanya menggunakan sebuah fasilitas pencarian. Sistem Temu Kembali Informasi atau yang biasa disebut *Information Retrieval* dimana seorang pengguna memungkinkan dapat menemukan dokumen yang diinginkan. *Information Retrieval* merupakan suatu sistem untuk melakukan pencarian, penyimpanan, dan juga pemeliharaan informasi. Informasi yang dimaksudkan dalam hal ini salah satunya terdiri dari teks (Hasanah, 2017). *Information Retrieval* bekerja dengan cara memisahkan masing-masing dokumen yang mungkin saling berhubungan dari sekumpulan dokumen-dokumen yang sudah ada sebelumnya.

Namun, sebelum mengembalikan hasil pencarian kepada pengguna, *Information Retrieval* melakukan beberapa proses. Salah satu proses penting yang dilakukan adalah dengan melakukan transformasi

terhadap semua kata pada dokumen-dokumen teks dan mengubahnya menjadi kata dasar, biasanya disebut dengan *stemming* (Anugrah, 2018). *Stemming* adalah suatu rangkaian tindakan untuk memperoleh kata dasar (*root* atau *stem*) dari masing-masing kata dengan mengeliminasi seluruh imbuhan (*affixes*) yang terdapat dalam kata tersebut baik itu berupa sisipan (*suffixes*), awalan (*prefixes*), serta kombinasi antara awalan dan akhiran (*confixes*) pada kata turunan (Wahyudi, 2017). *Stemming* merupakan pusat dari teknik pemrosesan *natural language* untuk memperoleh informasi kembali (*Information Retrieval*) yang efektif dan efisien (Frakes, 1992).

Setiap algoritma *stemming* mempunyai kelebihan dan kekurangan. Secara umum, untuk mengukur keefektifitasan dari sebuah algoritma *stemming* dapat didasarkan pada berbagai parameter yaitu keakuratan, kecepatan proses, dan juga kesalahan (Herlingga, 2020). Algoritma *stemming* bisa berbeda-beda, tergantung bahasa yang digunakan. Hal ini dikarenakan oleh morfologi dari bahasa-bahasa tersebut berbeda-beda (Wibawa, 2020). Contohnya, perbedaan morfologi Bahasa Indonesia dengan Bahasa Inggris yang menyebabkan algoritma *stemming* juga berbeda pada dua bahasa tersebut (Wahyudi, 2017). Jika dibandingkan proses *stemming* antara bahasa Indonesia dan bahasa

Inggris, tentu proses *stemming* bahasa Indonesia lebih rumit. Hal ini dikarenakan untuk memperoleh kata dasar yang diinginkan dari sebuah kata pada bahasa Indonesia harus mengeliminasi banyak jenis imbuhan (Agusta, 2009).

Terdapat beberapa penelitian yang telah dilakukan pada analisis perbandingan algoritma *stemming*. Diantaranya, penelitian yang dilakukan oleh Muhammad Latif Rozi, dkk ditemukan bahwa algoritma Vega memiliki waktu proses yang lebih kecil dengan waktu proses rata-rata 0,04 detik sedangkan untuk akurasi, algoritma Nazief Adriani diperoleh nilai akurasi yang lebih besar dengan nilai rata-rata sebesar 82,84% (Rozi, 2013). Penelitian lainnya juga dilakukan oleh Dwi Wahyudi, dkk melakukan uji perbandingan pada dokumen teks berbahasa Indonesia antara algoritma *stemming* Nazief Adriani dan juga algoritma *stemming* Porter. Hasil yang didapatkan adalah waktu proses algoritma Nazief Adriani hampir 2 kali lipat lebih lama dibandingkan dengan algoritma Porter. Sedangkan hasil terbaik pada pengukuran akurasi, algoritma Nazief Adriani unggul dengan tingkat akurasi sebesar 95,26 % (Wahyudi, 2017).

Penelitian ini akan membahas mengenai perbandingan algoritma *stemming* antara algoritma Arifin Setiono dengan algoritma Nazief Adriani. Algoritma Arifin Setiono dipilih sebagai perbandingan karena algoritma tersebut mempunyai tingkat akurasi yang relatif cukup tinggi (Simarankir, 2017). Untuk itu akan dilakukan analisa algoritma *stemming* untuk menentukan algoritma mana yang mempunyai akurasi serta waktu proses yang lebih baik dalam menentukan kata dasar pada sebuah dokumen bahasa Indonesia.

2. RUANG LINGKUP

Cakupan permasalahan yang akan dibahas pada penelitian ini yaitu membandingkan algoritma *stemming* Arifin Setiono dan Algoritma Nazief Adriani pada dokumen yang ditulis dalam bahasa Indonesia untuk mengukur performansi yang lebih baik yang dilihat dari akurasi dan kecepatan waktu proses dari kedua algoritma tersebut.

Pada penelitian ini, batasan-batasan permasalahannya adalah yang pertama, dokumen teks yang digunakan merupakan dokumen teks dalam bahasa Indonesia yang diambil pada media internet. Kedua, jumlah daftar *stopwords* sebanyak 758 kata. Ini diambil dari penelitian sebelumnya (Tala, 2004). Tidak ada penambahan *stopwords*. Ketiga, kamus kata dasar menggunakan sebanyak 28.526 kata dasar. Kamus kata dasar tidak ditambahkan. Dan keempat, jumlah kata pada dokumen dibatasi sebanyak 5500 kata.

Rencana hasil yang diinginkan dari pembahasan penelitian ini yaitu perbandingan tingkat akurasi dan kecepatan waktu proses *stemming* pada algoritma Nazief Adriani dan algoritma Arifin Setiono.

3. BAHAN DAN METODE

Bahan dan metode yang digunakan pada penelitian ini akan dijelaskan pada bab ini.

3.1 Information Retrieval

Pada dasarnya, *information retrieval* adalah proses yang dapat digunakan untuk menentukan dokumen yang berada dalam koleksi yang ditemubalikkan informasi sesuai dengan keinginan pengguna (Wahyudi, 2017).

Information Retrieval merupakan sebuah sistem yang mempunyai kemampuan dalam mencari, menyimpan, dan juga memelihara informasi. Konteks informasi yang dimaksud di sini adalah berupa teks (baik berupa tanggal dan numerik), video, audio, gambar, dan objek multimedia lainnya. (Kowalski, 2000).

3.2 Stemming

Proses *stemming* merupakan salah satu pra-pemrosesan teks yang mengembalikan kata ke bentuk semula bentuk dasar. Sebagian besar kata yang berasal dari imbuhan (*prefix, suffix, of infix*) mengandung arti teks. Oleh karena itu, untuk mempertahankan makna teks, proses *stemming* penting dilakukan diadakan. Pada dasarnya, algoritma *stemming* dimulai dari algoritma Porter untuk bahasa Inggris. Uniknya, setiap bahasa memiliki karakteristik dan struktur, khususnya struktur afiks, sehingga algoritma *stemming* akan ditransformasikan menjadi sesuai dengan karakteristik bahasa (Jumadi, 2021).

3.3 Algoritma Stemming Nazief-Adriani

Algoritma *stemming* untuk dokumen teks dalam bahasa Indonesia yang banyak digunakan dan populer adalah algoritma Nazief-Adriani (Jumadi, 2021). Tahapan-tahapan algoritma *stemming* yang diperkenalkan oleh Nazief dan Adriani adalah berikut ini: (Nugroho, 2017)

1. Temukan kata dalam kamus yang hendak dilakukan *stem*. Jika ada kata yang sama ditemukan dalam kamus, maka diduga kata tersebut sebagai *root word* dan algoritma akan berhenti.
2. Membuang *Inflection Suffixes* ("-kah", "-ku", "-lah", "-nya", atau "-mu"). Jika ternyata berupa *particles* ("-pun", "-tah", "-kah" atau "-lah") maka ulangi lagi *step* tersebut untuk melakukan penghapusan *Possesive Pronouns* ("-mu", "-nya", atau "-ku"), jika ada.
3. Menghilangkan *Derivation Suffixes* ("-kan", "-i" atau "-an"). Algoritma akan berhenti, apabila di dalam kamus terdapat kata. Tetapi, jika tidak ada, lanjutkan ke *step* 3a.
 - 1) Jika ternyata "-an" sudah terhapus dan kata dengan huruf terakhir yang dimaksudkan adalah "-k", maka hapus juga huruf terakhir "-k". Algoritma akan berhenti, apabila di dalam kamus terdapat kata. Tetapi, jika kata

itu tidak ada ditemukan dalam kamus maka akan dilakukan langkah 3b.

- 2) Akhiran yang dihapus ("-kan", "-i" atau "-an") lalu dikembalikan, setelah itu lanjutkan ke *step* 4.
4. Menghapus *Derivation Prefix*. Apabila pada *step* sebelumnya atau ada sufiks yang telah dihapus pada langkah 3, maka langkah selanjutnya menuju ke *step* 4a. Tetapi, jika tidak, lanjutkan ke *step* 4b.
 - 1) Lakukan pemeriksaan terhadap tabel yang berisi kombinasi awalan-akhiran yang tidak diizinkan. Algoritma akan berhenti, apabila ditemukan. Tetapi, jika tidak, lanjutkan ke *step* 4b.
 - 2) For $i = 1$ to 3, jenis awalan harus sudah ditentukan lalu awalan dihapus. Jika ternyata belum juga ditemukan *root word*, maka lanjutkan ke *step* 5. Tetapi, algoritma akan berhenti, apabila ditemukan. Catatan: jika awalan kedua dan awalannya sama maka algoritma berhenti.
5. Lakukan Perekamannya (*Recording*).
6. Apabila seluruh proses sudah dilakukan dan sudah selesai, namun belum berhasil juga, maka dapat diasumsikan bahwa *root word*-nya adalah kata awal tersebut. Lalu seluruh proses pun dapat dinyatakan selesai..

3.4 Algoritma Stemming Nazief-Adriani

Pada algoritma ini akan didahului dengan membaca setiap kata yang ada. Langkah-langkah selanjutnya yang terjadi pada algoritma ini adalah: (Novitasari, 2016)

1. Memeriksa seluruh bentuk kata yang memungkinkan. Untuk masing-masing kata yang telah diperiksa dapat diduga bahwa kata terdiri dari dua Awalan (*prefiks*) dan tiga Akhiran (*sufiks*). Apabila ternyata pada saat pemeriksaan kata tersebut tidak mempunyai imbuhan seperti banyaknya imbuhan layaknya formula yang sudah dijelaskan di atas, maka awalan (*prefiks*) akan diberi tanda x dan akhiran (*sufiks*) akan diberikan tanda xx pada imbuhan yang ada atau kosong.
2. Secara berurutan akan dilakukan pemenggalan dalam Algoritma ini yaitu:
AW : AW (Awalan)
AK : AK (Akhiran)
KD : KD (Kata Dasar)
 - 1) AW I, hasil pemenggalan kata akan disimpan dalam pe_1 (*prefiks* 1)
 - 2) AW II hasil pemenggalan kata akan disimpan dalam pe_2 (*prefiks* 2)
 - 3) AK I hasil pemenggalan kata akan disimpan dalam su_1 (*sufiks* 1)
 - 4) AK II, hasil pemenggalan kata akan disimpan dalam su_2 (*sufiks* 2)
 - 5) AK III, hasil pemenggalan kata akan disimpan dalam su_3 (*sufiks* 3)

Untuk masing-masing langkah pemenggalan kata yang telah dilakukan seperti penjelasan di atas biasanya dilanjutkan dengan proses pengecekan kata ke kamus. Hal tersebut untuk memastikan apakah kata yang dipotong hasilnya sudah sesuai dengan bentuk dasar atau tidak. Apabila proses ini dinyatakan berhasil maka tahapan ini selesai dan proses selanjutnya yaitu pemenggalan imbuhan tidak perlu dilanjutkan lagi.

3. Namun, jika tetap juga tidak ditemukan di dalam kamus, padahal proses pemenggalan sudah dilakukan sampai AK III, maka proses kombinasi adalah proses yang akan dilakukan selanjutnya. Di mana, hasil pemenggalan kata tersebut yaitu kata dasarnya akan dikombinasikan bersama dengan imbuhan dengan 12 formula berikut ini:
 - 1) KD
 - 2) KD + AK III
 - 3) KD + AK III + AK II
 - 4) KD + AK III + AK II + AK I
 - 5) AW I + AW II + KD
 - 6) AW I + AW II + KD + AK III
 - 7) AW I + AW II + KD + AK III + AK II
 - 8) AW I + AW II + KD + AK III + AK II + AK I
 - 9) AW II + KD
 - 10) AW II + KD + AK III
 - 11) AW II + KD + AK III + AK II
 - 12) AW II + KD + AK III + AK II + AK I

Untuk kombinasi a, b, c, d, h, dan l sudah dilakukan pemeriksaan pada langkah sebelumnya, karena itu merupakan kombinasi dari hasil proses pemenggalan secara bertahap itu. Jadi, tinggal 6 kombinasi lagi yang perlu dikerjakan yaitu kombinasi e, f, g, i, j, dan k. Jika proses-proses kombinasi yang dikerjakan itu ternyata ada, maka tidak diperlukan lagi proses pemeriksaan pada kombinasi yang lainnya.

Dibutuhkan proses terhadap 12 kombinasi di atas, karena bisa saja terjadi fenomena *oversteaming* pada algoritma proses pemenggalan imbuhan. Hal ini merupakan kelemahan yang mengakibatkan pemenggalan bagian kata yang pada dasarnya merupakan milik dari kata dasar itu sendiri yang mungkin menyerupai imbuhan sejenis yang ada. Demikian halnya untuk 12 kombinasi itu, pemenggalan kata yang telah terlanjur terjadi, dapat dilakukan pengembalian kata sesuai dengan posisinya.

3.5 Analisis Kebutuhan Data

Tahapan awal dari analisis sistem adalah analisis kebutuhan data. Data-data yang diinputkan itu adalah:

1. Data dokumen teks berbahasa Indonesia diambil dari media internet dengan cara melakukan pencarian dokumen teks dengan berbagai jenis kategori dokumen. Isi dokumen teks tersebut

kemudian disalin ke dalam *notepad* untuk dijadikan sebagai data testing.

2. Jumlah daftar *stopwords* sebanyak 758 kata. Ini diambil dari penelitian sebelumnya (Tala, 2004).
3. Data berupa kumpulan kata dasar dalam dokumen teks bahasa Indonesia yang digunakan untuk proses *stemming* merupakan daftar kata dasar yang dijadikan sebagai kamus. Kamus kata dasar berbahasa Indonesia ini dibatasi jumlahnya sebanyak 28.526 kata dasar.

3.6 Metode pengujian

Untuk menguji sistem yang sudah dibuat pada penelitian ini digunakan data uji yang sudah dibuat sebelumnya. Pengujian yang dilakukan untuk mengukur akurasi dari kedua algoritma yaitu membandingkan bentuk kata hasil proses *stemming* dengan kata yang ada dalam kamus dan juga untuk mengukur kecepatan kedua algoritma terkait dengan waktu eksekusi (*execution time*) pada saat proses eksekusi berlangsung.

Di bawah ini adalah formula (1) yang digunakan untuk mengukur akurasi (Wahyudi, 2017) yaitu:

$$\text{Akurasi} = \frac{\text{Jumlah term yang benar}}{\text{Jumlah term hasil stemming}} \times 100\% \quad (1)$$

4. PEMBAHASAN

Bagian ini dijelaskan tentang tahapan yang dilakukan pada penelitian, diantaranya:

4.1 Pengujian Sistem

Proses pengujian algoritma dapat dilihat dengan menganalisis nilai yang dihasilkan yaitu durasi pemrosesan dan akurasi yang diperoleh dari inputan banyaknya jumlah kata pada dokumen teks dengan hasil proses *stemming* yang tidak sesuai.

4.1.1 Skenario Pengujian

Skenario pengujian dilakukan dengan menguji sebanyak 30 dokumen teks di mana jumlah kata pada masing-masing dokumen berbeda-beda. Hasil keluaran yang diharapkan berupa teks hasil proses *stemming*, kemudian proses selanjutnya adalah menghitung jumlah kata hasil proses *stemming* yang tidak sesuai dari kedua algoritma *stemming* tersebut.

4.1.2 Pengujian

Pengujian ini dilakukan dengan meninjau dua algoritma berbeda yang menggunakan dokumen dengan ukuran jumlah kata yang sama. Algoritma pertama yang akan diuji adalah algoritma Nazief Adriani lalu kemudian algoritma Arifin Setiono. Langkah pengujian ini dilakukan untuk mengukur kecepatan serta menganalisis ketetapan (akurasi) masing-masing algoritma dalam menemukan kata dasar dari sebuah teks di dalam dokumen.

Untuk menguji kecepatan algoritma dilakukan dengan cara mengukur waktu eksekusi (*execution time*) algoritma pada saat melakukan proses *stemming* dengan menggunakan satuan *microtime*. Sementara pengujian akurasi algoritma dilakukan dengan menghitung jumlah kata yang berhasil di *stem* dengan benar lalu dibagi dengan jumlah kata yang berhasil di *stem* kemudian di kali dengan seratus.

4.2 Hasil Pengujian

Setelah dilakukan pengujian terhadap 30 dokumen teks berbahasa Indonesia, maka dilakukan analisa hasil pengujian untuk kedua algoritma *stemming* yang ditampilkan pada penjelasan berikut.

4.2.1 Hasil Pengujian Algoritma Nazief & Adriani

Hasil pengujian algoritma *stemming* Nazief Adriani terhadap 30 dokumen teks berbahasa Indonesia yang dapat kita lihat pada tabel 1.

Tabel 1. Hasil pengujian algoritma Nazief Adriani

No.	Dokumen Teks	Jumlah Kata	Waktu Proses (det)	Akurasi (%)
1	algoritma.txt	257	8.37	99.35
2	antarmuka_grafis.txt	58	1.11	100
3	bluemoon.txt	263	2.73	98.95
4	ergonomi.txt	475	13.27	96.48
5	forensik.txt	600	16.81	97.61
6	garis_waktu.txt	350	9.34	96.09
7	gubernur_jambi.txt	326	14.05	98.65
8	habibie.txt	810	21.54	97.62
9	it.txt	195	5.09	99.03
10	jendela_dunia.txt	128	1.63	100
11	jenis_sampah.txt	227	3.97	96.04
12	kalau_cinta_melekat.txt	1157	22.41	99.37
13	latar_belakang.txt	301	9.9	97.16
14	legenda_telaga_warna.txt	569	12.33	97.99
15	keledai_membaca.txt	180	4.05	100
16	motivasi_belajar.txt	1488	40.94	97.15
17	nyanyian.txt	660	19.39	97.80
18	peduli_lingkungan.txt	186	3.56	99.06
19	pengantar.txt	381	7.92	97.09
20	perpercayaan_padaku.txt	5371	141.05	97.16
21	professional.txt	2269	50.42	98.40
22	revolusi_teknologi.txt	315	7.25	95.73
23	ronaldo.txt	1385	25.36	99.03
24	senjata_menulis_buku.txt	504	14.84	97.85
25	sepakbola.txt	214	4.72	97.25
26	setelah_berpikir.txt	1508	26.34	93.60
27	soekarno.txt	2383	69.23	97.48
28	suatu_malam.txt	274	5.47	95.65
29	teluk_jailolo.txt	1297	28.02	96.27
30	touchscreen.txt	611	13.93	98.12

Berdasarkan tabel 1 kita dapat melihat hasil uji terhadap 30 dokumen, di mana ditemukan 3 hasil pengujian yang memiliki nilai akurasi mencapai 100%.

Pengujian itu dilakukan pada dokumen antarmuka_grafis.txt dengan jumlah kata 257, dokumen jendela_dunia.txt dengan jumlah kata 128, serta dokumen keledai_membaca.txt dengan jumlah kata 180.

4.2.2 Hasil Pengujian Algoritma Arifin Setiono

Berikut ini adalah hasil pengujian algoritma *stemming* Arifin Setiono terhadap 30 dokumen teks berbahasa Indonesia yang dapat dilihat pada tabel 2.

Tabel 2. Hasil pengujian algoritma Arifin Setiono

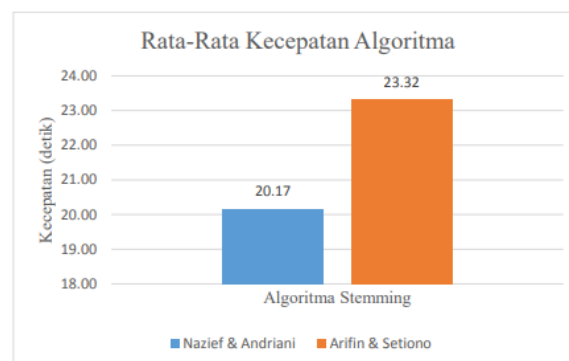
No.	Dokumen Teks	Jumlah Kata	Waktu Proses (det)	Akurasi (%)
1	algoritma.txt	257	13.24	96.13
2	antarmuka_grafis.txt	58	0.7	100
3	bluemoon.txt	263	5.32	96.84
4	ergonomi.txt	475	14.74	92.58
5	forensik.txt	600	20.29	93.13
6	garis_waktu.txt	350	8.56	96.65
7	gubernur_jambi.txt	326	22.46	93.72
8	habibie.txt	810	34.46	94.59
9	it.txt	195	3.49	93.20
10	jendela_dunia.txt	128	1.17	94.34
11	jenis_sampah.txt	227	3.76	94.06
12	kalau_cinta_melekat.txt	1157	22.17	95.14
13	latar_belakang.txt	301	8.19	94.89
14	legenda_telaga_warna.txt	569	10.14	94.31
15	keledai_membaca.txt	180	5.2	100
16	motivasi_belajar.txt	1488	43.45	92.39
17	nyanyian.txt	660	23.99	90.11
18	peduli_lingkungan.txt	186	2.9	98.11
19	pengantar.txt	381	5.92	97.77
20	percayakan_padaku.txt	5371	150.25	94.79
21	professional.txt	2269	48.17	93.88
22	revolusi_teknologi.txt	315	8.41	94.51
23	ronaldo.txt	1385	30.55	91.62
24	senjata_menulis_buku.txt	504	10.28	96.57
25	sepakbola.txt	214	5.3	88.99
26	setelah_berpikir.txt	1508	31.7	93.31
27	soekarno.txt	2383	102.42	93.89
28	suatu_malam.txt	274	4.93	89.57
29	teluk_jailolo.txt	1297	37.23	94.10
30	touchscreen.txt	611	20.3	94.98

Berdasarkan tabel 2 kita dapat melihat hasil uji terhadap 30 dokumen, di mana ditemukan 2 hasil pengujian yang memiliki nilai akurasi mencapai 100%. Pengujian itu dilakukan pada dokumen antarmuka_grafis.txt dengan jumlah kata 257, serta dokumen keledai_membaca.txt dengan jumlah kata 180.

4.2.3 Hasil Rata-Rata Pengujian Algoritma

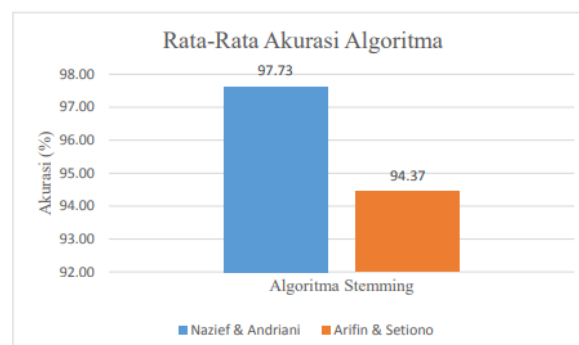
Berdasarkan hasil pengujian Nazief Adriani yang ditunjukkan pada tabel 1 dan algoritma Arifin Setiono yang ditunjukkan pada tabel 2, bila total waktu proses dan total akurasi untuk masing-masing algoritma dibagi dengan banyaknya data pengujian yakni 30 dokumen,

maka diperoleh nilai rata-rata waktu proses dan juga akurasi yang dapat dilihat pada gambar 1 dan gambar 2 berikut ini:



Gambar 1. Grafik hasil rata-rata waktu proses algoritma

Berdasarkan data pada gambar 1 ditunjukkan bahwa nilai waktu rata-rata yang paling kecil pada saat proses *stemming* berlangsung dimiliki oleh algoritma Nazief Adriani dengan waktu proses sebesar 20.17 detik sementara algoritma Arifin Setiono memiliki waktu proses rata-rata sebesar 23.32 detik.



Gambar 2. Grafik hasil rata-rata akurasi algoritma

Berdasarkan data grafik pada gambar 2 dapat terlihat bahwa nilai rata-rata yang paling besar terkait dengan akurasi *stemming* dimiliki oleh algoritma Nazief Adriani dengan nilai akurasi mencapai 97.73 % sementara algoritma Arifin Setiono memiliki rata-rata nilai akurasi mencapai 94.37%.

4.2.4 Analisa Kesalahan Hasil Stemming

Tingkat kesalahan yang dihasilkan saat proses *stemming* dilakukan pada kedua algoritma memiliki perbedaan yang diakibatkan oleh perbedaan karakteristik dari masing-masing algoritma ketika algoritma melakukan *stemming* pada setiap kata. Beberapa kesalahan yang ditemui adalah kesalahan dalam pengetikan kata, orang, adanya nama tempat, istilah, dan juga bahasa asing yang memiliki kesamaan term dalam aturan pemenggalan kata baik itu pada algoritma *stemming* yang menggunakan aturan imbuhan atau juga

algoritma yang menggunakan kamus. Berikut ini beberapa kesalahan hasil *stemming* yang ditemukan pada dokumen yang telah diuji.

1. Kesalahan *stemming* akibat kesalahan pada pengetikan kata yang dapat dilihat pada tabel 3 berikut ini:

Tabel 3. Kesalahan *stemming* akibat kesalahan pada pengetikan kata

No	Kata	Seharusnya	Hasil Stemming	Kata Dasar
1	teknisya	teknisnya	teknisya	teknis
2	penyelesaiannya	penyelesaiannya	penyelesaiannya	selesai
3	kecerahana	kecerahan	kecerahana	cerah
4	keamaan	keamanan	ama	aman
5	mengijinkan	mengizinkan	mengijinkan	izin
6	pengabungan	penggabungan	pengabungan	gabung
7	kepolisan	kepolisian	kepolisan	polisi
8	mapun	maupun	ma	mau
9	didupnya	hidupnya	dup	hidup
10	tidakan	tindakan	tidak	tindak
11	pengerak	penggerak	erak	gerak
12	mengulan	mengulang	ulan	ulang
13	jaman	zaman	jam	zaman
14	tingi	tinggi	tingi	tinggi
15	telfon	telepon	fon	telepon
16	eku	aku	e	aku
17	kenyatan	kenyataan	kenyat	nyata

2. Kesalahan *stemming* untuk nama tempat, istilah, dan nama orang yang dapat dilihat pada tabel 4 berikut ini;

Tabel 4. Kesalahan *stemming* untuk nama tempat, istilah, dan nama orang

No.	Kata	Hasil Stemming	Seharusnya
1	bali	bali	tetap
2	kuba	ba	tetap
3	maluku	malu	tetap
4	dimas	mas	tetap

3. Kesalahan *stemming* untuk bahasa asing yang dapat dilihat pada tabel 5 berikut ini

Tabel 5. Kesalahan *stemming* untuk bahasa asing

No.	Kata	Hasil Stemming	Seharusnya
1	diinputkan	diinputkan	bukan kata dasar bahasa asing
2	diposting	diposting	bukan kata dasar bahasa asing
3	die	e	tetap

Selain beberapa bentuk kesalahan yang telah diuraikan diatas, terdapat juga kesalahan yang umum ditemui pada sebuah algoritma *stemming*. Kesalahan tersebut yakni *Overstemming*, *Understemming*,

Unchange, dan *Spelling Exception*. Berikut beberapa kesalahan hasil *stemming* pada bentuk ini yang ditemukan pada dokumen yang telah diuji dan dapat dilihat pada tabel 6 berikut ini:

Tabel 6. Kesalahan hasil *stemming* karena pemenggalan kata

Term	Hasil Stemming	Seharusnya	Jenis Kesalahan
tersedia	dia	sedia	<i>overstemming</i>
kekurangannya	rang	kurang	<i>overstemming</i>
tangan	tang	tangan	<i>overstemming</i>
disebarkan	bar	sebar	<i>overstemming</i>
disebakan	bab	sebab	<i>overstemming</i>
membacakan	bacak	baca	<i>understemming</i>
disegani	segni	segan	<i>understemming</i>
terancam	rancam	ancam	<i>understemming</i>
diragukan	raguk	ragu	<i>understemming</i>
melarikan	larik	lari	<i>understemming</i>
memiliki	memiliki	milik	<i>unchange</i>
tindakan	tindakan	tindak	<i>unchange</i>
pemanfaatan	pemanfaatan	manfaat	<i>unchange</i>
kutuanakan	kutuanakan	tuang	<i>unchange</i>
mengoleksi	mengoleksi	koleksi	<i>unchange</i>
menangani	angan	tangan	<i>spelling exception</i>
peralatan	ralat	alat	<i>spelling exception</i>
pengukuran	kukur	ukur	<i>spelling exception</i>
perancangan	ancang	rancang	<i>spelling exception</i>
mengalami	kalam	alami	<i>spelling exception</i>

5. KESIMPULAN

Perbandingan performansi algoritma *stemming* Nazief Adriani lebih baik dari algoritma Arifin Setiono dengan rata-rata nilai akurasi mencapai 97,73 persen serta rata-rata waktu proses sebesar 20,17 detik. Sementara algoritma Arifin Setiono hanya memiliki rata-rata nilai akurasi sebesar 94,37 persen serta rata-rata waktu proses sebesar 23,32 detik. Akurasi juga dapat dipengaruhi oleh kesalahan *stemming* yang terjadi akibat kesalahan pengetikan kata, kesalahan *stemming* untuk nama tempat, istilah, dan nama orang, kesalahan *stemming* untuk bahasa asing, dan yang terakhir kesalahan *stemming* karena pemenggalan kata. Akurasi juga dapat dipengaruhi oleh kesalahan *stemming*.

6. SARAN

Perlu dilakukan modifikasi pada setiap algoritma agar dapat mengurangi tingkat kesalahan, yakni dengan menyempurnakan aturan pemenggalan imbuhan baik itu berupa partikel, awalan, akhiran, maupun sisipan.

Penggunaan kamus kata dasar mempengaruhi hasil akurasi *stemming*. Semakin besar jumlah kata dasar yang

digunakan pada kamus akan semakin baik juga hasil stemming yang didapatkan.

7. DAFTAR PUSTAKA

- Agusta, L. (2009), Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Adriani untuk Stemming Dokumen Teks Bahasa Indonesia. Konferensi Nasional Sistem dan Informatika, KNS&I09-036.
- Anugrah, I. G., Rosyid, H. (2018), Penerapan Information Retrieval Menggunakan Pemodelan Topik Pada Deskripsi Portal Multimedia. Jurnal Nasional Komputasi dan Teknologi Informasi, 2(1), 48-54.
- Frakes, W. B. (1992), Information Retrieval Data Structure and Algorithms. New Jersey: Prentice-Hall.
- Hasanah, N. (2017), Sistem Pencarian Skripsi Berbasis Information Retrieval di FASTIKOM UNSIQ. Jurnal PPKM, 1(1), 105-113.
- Herlingga, A. C., Prismana, P. E. IGL., Prehanto, D. R., Dermawan, D. A. (2020), Algoritma Stemming Nazief & Adriani Dengan Metode Cosine Similarity Untuk Chatbot Telegram Terintegrasi Dengan E-layanan. Journal of Informatics and Computer Science, 02(01), 19-26.
- Jumadi, J., Maylawati, D. S., Pratiwi1, L. D, M., Ramdhani, A. (2021), Comparison of Nazief-Adriani and Paice-Husk algorithm for Indonesian text stemming process. The 5th Annual Applied Science and Engineering Conference (AASEC 2020), 1098, IOP Publishing. <https://doi:10.1088/1757-899X/1098/3/032044>
- Kowalski, G. J. (2000), Information Storage and Retrieval systems: Theory and Implementation. United States.
- Novitasari, D. (2016), Perbandingan Algoritma Stemming Porter Dengan Arifin Setiono Untuk Menentukan Tingkat Ketepatan Kata Dasar. Jurnal String, 1(2), 120-129.
- Nugroho, H. T. (2017), Pengaruh Algoritma Stemming Nazief-Adriani Terhadap Kinerja Algoritma Winnowing Untuk Mendeteksi Plagiarisme Bahasa Indonesia. ULTIMA COMPUTING: JURNAL SISTEM KOMPUTER, IX(1), 36-40.
- Rozi, M. L. (2013), Implementasi dan Analisis Perbandingan Algoritma Stemming Nazief & Adriani dengan Algoritma Stemming Vega dalam Information Retrieval System. Universitas Telkom, Bandung.
- Simarankir, M. S. H. (2017), "Studi Perbandingan Algoritma-Algoritma Stemming untuk Dokumen Teks Bahasa Indonesia. Jurnal Inkofar, 1(1), 41-47.
- Tala F. Z. (2004), A Study of Stemming Effect on Information Retrieval in Bahasa Indonesia. Netherland, Universiteit van Amsterdam
- Wahyudi, D., Susyanto, T., Nugroho, D. (2017), Implementasi Dan Analisis Algoritma Stemming Nazief & Adriani Dan Porter Pada Dokumen Berbahasa Indonesia. Jurnal Ilmiah SINUS, 15(2), 49-56.
- Wibawa, A. P., Dwiyanto, F. A., Zaeni, I. A. E, Nurrohman, R. K., Afandi, AN. (2020), Stemming javanese affix words using Nazief and Adriani modifications. JURNAL INFORMATIKA, 14(1), 36-42.